1 - Kapsack

0-1 Knapsack problem: dynamic programming approach

- We can do better with an algorithm based on dynamic programming
- We need to carefully identify the subproblems

8

Defining a Subproblem

- Given a knapsack with maximum capacity W, and a set S consisting of n items
- Each item i has some weight w_i and benefit value b_i (all w_i and W are integer values)
- Problem: How to pack the knapsack to achieve maximum total value of packed items?

9

د/ [illegible]
0508793242

Defining a Subproblem

- We can do better with an algorithm based on dynamic programming

- We need to carefully identify the subproblems

Let's try this:
If items are labeled *1..n*, then a subproblem would be to find an optimal solution for
S_k = *{items labeled 1, 2, .. k}*

Defining a Subproblem

If items are labeled *1..n*, then a subproblem would be to find an optimal solution for S_k = *{items labeled 1, 2, .. k}*

- This is a reasonable subproblem definition.
- The question is: can we describe the final solution (S_n) in terms of subproblems (S_k)?
- Unfortunately, we <u>can't</u> do that.

3

Defining a Subproblem

$w_1=2$ $b_1=3$	$w_2=4$ $b_2=5$	$w_3=5$ $b_3=8$	$w_4=3$ $b_4=4$?	

Max weight: W = 20

For S_4:

Total weight: 14

Maximum benefit: 20

Item #	Weight w_i	Benefit b_i
1	2	3
2	4	5
3	5	8
4	3	4
5	9	10

S_4 S_5

$w_1=2$ $b_1=3$	$w_2=4$ $b_2=5$	$w_3=5$ $b_3=8$	$w_5=9$ $b_5=10$

For S_5:

Total weight: 20

Maximum benefit: 26

Solution for S_4 is not part of the solution for S_5!!!

Defining a Subproblem

- As we have seen, the solution for S_4 is not part of the solution for S_5
- So our definition of a subproblem is flawed and we need another one!

4

Defining a Subproblem

- Given a knapsack with maximum capacity W, and a set S consisting of n items
- Each item i has some weight w_i and benefit value b_i (all w_i and W are integer values)
- Problem: How to pack the knapsack to achieve maximum total value of packed items?

Defining a Subproblem

- Let's add another parameter: w, which will represent the maximum weight for each subset of items
- The subproblem then will be to compute $V[k,w]$, *i.e.*, to find an optimal solution for S_k = *{items labeled 1, 2, .. k} in a knapsack of size w*

5

Running time

```
for w = 0 to W
    V[0,w] = 0
for i = 1 to n
    V[i,0] = 0
for i = 1 to n
    for w = 0 to W
        < the rest of the code >
```

$O(W)$

Repeat n times

$O(W)$

What is the running time of this algorithm?

O(n*W)

Remember that the brute-force algorithm takes $O(2^n)$

يوضع جميع المجموعات الجزئية

ويعمل check لـ

20

Example

Let's run our algorithm on the following data:

n = 4 (# of elements)
W = 5 (max weight)
Elements (weight, benefit):
(2,3), (3,4), (4,5), (5,6)

21

	(')	1	2	3	4
wk=	w_i	2	3	4	5
bk	b_i	3	4	5	6

6

Recursive Formula

$$V[k,w] = \begin{cases} V[k-1,w] & \text{if } w_k > w \\ \max\{V[k-1,w], V[k-1,w-w_k]+b_k\} & \text{else} \end{cases}$$

- The best subset of S_k that has the total weight $\leq w$, either contains item k or not.
- First case: $w_k > w$. Item k can't be part of the solution, since if it was, the total weight would be $> w$, which is unacceptable.
- Second case: $w_k \leq w$. Then the item k can be in the solution, and we choose *the case with greater value.*

max weight

items // nw

$O(nw)$

0-1 Knapsack Algorithm

```
for w = 0 to W
    V[0,w] = 0
for i = 1 to n
    V[i,0] = 0
for i = 1 to n
    for w = 0 to W
        if w_i <= w // item i can be part of the solution
            if b_i + V[i-1,w-w_i] > V[i-1,w]
                V[i,w] = b_i + V[i-1,w- w_i]
            else
                V[i,w] = V[i-1,w]
        else V[i,w] = V[i-1,w]  // w_i > w
```

max weight

max

7

$v(k,w)$

$v(1,5)$

$w = 5$

$k=1 \Rightarrow w_k = w_1 = 2$

w_k b_k

Finding the Items (3)

w', y'

Items: w b
1: (2,3)
2: (3,4)
3: (4,5)
4: (5,6)

K ← i\W

i\W	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	3	3	3	3
2	0	0	3	4	4	7
3	0	0	3	4	5	7
4	0	0	3	4	5	7

item ← i

i=3
k= 5
b_i=5
w_i=4
$V[i,k] = 7$
$V[i-1,k] = 7$

```
i=n, k=W
while i,k > 0
    if V[i,k] ≠ V[i−1,k] then
        mark the i^th item as in the knapsack
        i = i−1, k = k-w_i
    else
        i = i−1
```


Finding the Items (4)

Items: w b
1: (2,3)
2: (3,4)
3: (4,5)
4: (5,6)

i\W	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	3	3	3	3
2	0	0	3	4	4	7
3	0	0	3	4	5	7
4	0	0	3	4	5	7

i=2
k= 5
b_i=4
w_i=3
$V[i,k] = 7$
$V[i-1,k] = 3$
$k - w_i = 2$

$5 - w_2 = 2$

```
i=n, k=W
while i,k > 0
    if V[i,k] ≠ V[i−1,k] then
        mark the i^th item as in the knapsack
        i = i−1, k = k-w_i
    else
        i = i−1
```

$i = i - 1$

~~7~~ 8

$$[k, w] = \begin{cases} V[k-1, w] & \text{if } w_k > w \\ \max\{ V[k-1, w],\ V[k-1, w-w_k] + b_k \} \end{cases}$$

القانون

1 item w_k

$w_k = 2 > w = 1$

$V[1, 1] = V[0, 1] = 0$

k

?

$V[1, 2] = ??$

$w_k = w_1 = 2$

$w = 2$

$w_k \not> w$

$k = 1$

$$\max \begin{cases} V[0, 2] = 0 \\ V[0, 0] + 3 = 0 + 3 = 3 \end{cases}$$

$b_k = b_1$

$V[1, 2] = 3$

irst row (9) ~~i~~ item1 $\boxed{K=1}$

$V[K, w]$

$V[1, w] = ??$ $\quad w: 0, 1, 2, 3, 4, 5$

$w_k = 2 > 0, 1$

$\therefore \; V[1, 0] = V[0, 0] = 0$

$V[1, 1] = V[0, 1] = 0$

$V[1, w]$, $\quad w = 2, 3, 4, 5$

نعملها عادي ال max دي

$w_k > w$. false

$2 > 3, 4, 5$ false

$$V[\underset{\downarrow w}{1, 2}] = \max \begin{cases} V[0, 2] = 0 \\ V[0, 0] + b_1 = 0 + 3 = 3 \end{cases}$$

$$V[1, 3] = \max \begin{cases} V[0, 3] = 0 \\ V[0, 1] + b_1 = 0 + 3 = 3 \end{cases}$$

$$V[1, 4] = \max \begin{cases} V[0, 4] = 0 \\ V[0, 2] + b_1 = 0 + 3 = 3 \end{cases}$$

cond row (10) item 2 $\boxed{k=2}$

$V[k, w]$

$V[2, w] = ??$ $w : 0, 1, 2, 3, 4, 5$

$w_k = w_2 = 3 > 0, 1, 2,$

$\therefore V[2,0] = V[1,0] = 0$

$V[2,1] = V[1,1] = 0$

$V[2,2] = V[1,2] = 3$

...

باقي العناصر كما يلي نجد لها max لأنه

$w_k = w_2 = 3 > 3, 4, 5$ false

$$V[2, \underset{w}{3}] = \max \begin{cases} V[1,3] = 3 \\ V[1,0] + b_2 = 0 + 4 = 4 \end{cases}$$

$$V[2,4] = \max \begin{cases} V[1,4] = 3 \\ V[1,1] + b_2 = 0 + 4 = 4 \end{cases}$$

$$V[2,5] = \max \begin{cases} V[1,5] = 3 \\ V[1,2] + b_2 = 3 + 4 = 7 \end{cases}$$

0508793242

$V(3,4)$ — $k=3$, $w_k = w_3 = 4$, $b_k = 5$

$w = 4$

$\boxed{w_k \geq w \quad \checkmark}$

$k = 4$

$$V(3,4) = \max \begin{cases} V(2,4) = 4 \leftarrow \text{مش بحرول} \\ V(2,0) + b_k = 0 + 5 = 5 \end{cases}$$

$\therefore V(3,4) = 5$

$V(3,5)$, $k=3$, $w_k = w_3 = 4$, $b_k = 5$

$w = 5$ $\quad w_k \not> w$

$$V(3,5) = \begin{cases} V(2,5) = 7 \\ V(2,1) + 5 = 0 + 5 = 5 \end{cases}$$

(12)

~~3~~

$k = 4$

4th row

$k = 4$
$w_k = 5$
$w = 1$

$v[4, 1] =$

⋮

$k = 4$
$w_k = w_4 = 5$
$w = 5$

$v[4, 5] =$

$$\max = \begin{cases} v[3, 5] = 7 \\ v[3, 0] + b_4 = 0 + 6 = 6 \end{cases}$$

$v[4, 5] = 7$

$\boxed{\text{optimal benefit} = 7}$

2)

	1	2	3	4
w_i	2	3	4	5
b_i	3	4	5	6

ارسل يا شحمه

0508793242 | م/ [illegible]

	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	3	3	3	3
2	0	0	3	4	4	7
3	0	0	3	4	$\cancel{7}$ 5	7
4	0	0	3	4	5	7

$$\overset{K}{V}[1,\overset{w}{1}] = \begin{cases} V[0,1] = 0 \\ \end{cases}$$

$$V[1,2] = \begin{cases} \cancel{V[K-1,w] = V[0,2]} \end{cases}$$

$$\max(V[0,2] = 0,\ V[0,0] + 3 = 3)$$

$$V[1,3] = \max(V[0,3],\ V[0,\overset{1}{3-2}]+3 = 3$$

$$V[1,4] = \max(V[0,4],\ V[0,\overset{1}{4-3}]+3 = 3$$

$$V[1,5] = \max(V[0,5],\ V[0,\overset{3}{5-2}]+3 = 3$$

$$V[2,1] = V[1,1] = 0$$

$$V[2,3] = \max(V[1,\underset{3}{3}],\ V[1,\underset{0}{3-3}]+4) + 4 = 4$$

$$V[2,4] = \max(V[1,\underset{7}{4}],\ V[1,\overset{1}{\underset{0}{4-3}}]+4$$

$$V[2,5] = \cancel{\max}(\max(V[1,\underset{3}{5}],\ V[1,\underset{7}{5-3}]^{3}+4$$

$$V[2,2] = V[1,2] = 3$$

$$V[3,1] = V[2,1] = 0$$

$$V[3,4] = \max(V[2,4],\ V[2,\overset{0}{4-4}]+5 = 5$$

$$V[3,5] = \max(V[2,5],\ V[2,\overset{1}{\underset{0}{5-4}}]+5 = 7$$

$$V[4,5] = \max(V[\underset{7}{3},5],\ V[3,\underset{0}{5-5}]+6 = 7$$

لايجاد الحل كالتالي 14

والتوضيح التفصيلي في الصفحات التالية

$V[4,5] = 7$

$i = 4 \quad , \quad k = 5$

$V[4,5] \neq V[3,5]$

$i = 2$

$i = 1$

$i = 3$

$i = 2$

$k = k - w_2$

$k = 5 - 3 = k = 2$

$i = 1 , \quad k = 2$

$V[1,2] \neq V[0,2]$

$k = 2 - w_1 = 2 - 2 = 0$

$i = 0$

(15)

~~6~~ 7

~~ايش هي العناصر items~~

$i=4$, $k=5$

$V[4,5] = V[3,5] = 7$

$i = 4$ ~~x~~

~~$i=3$, $k=5-w_4=5-5=0$~~

$i = i-1 = 3$ اصعد صف

$i=3$

$k=5$

$V[3,5] = V[2,5] = 7$

$i=3$ ~~x~~

$i = i-1 = 2$ اصعد صف

$i=2$

$k=5$

$V[2,5] \neq V[1,5]$

$\boxed{i=2 \text{ is in solution}}$

اصعد صف

$i = i-1 = 1$

جديد $k = k - w_2 = 5-3 = 2$

$k = 2$

$i=1$

$k=2$

$V[1,2] \neq V[0,2]$

$\boxed{i=1 \text{ is in solution}}$

$i = i-1 = 0$ توقف

$k = k - w_1 =$

$k = 2-2 = \boxed{0}$ توقف

$\therefore$ The solution is $\{it_1, it_2\}$

Benifit $= 3 + 4 = \boxed{7}$

weights $= 2+3 = 5$

ملاحظة (16')

$$V[k,w] = \begin{cases} V[k-1, w] & \text{if } w_k > w \\ \max\{V[k-1, w], V[k-1, w-w_k] + b_k\} & \text{else} \end{cases}$$

$V[0, w] = 0$

$V[i, 0] = 0$

---

items which gives this benifit لاستخراج الحلول

---

if($B[i,k] \neq B[i-1, k]$)

mark ith item as in the knapsack

~~i = i-1~~ , $\boxed{k = k - w_i}$ $i = i - 1$

اصعد صف لأعلى و نقص الوزن

else

$i = i - 1$ ⟵ اصعد صف لأعلى

	1	2	3	4	5
w_k	2	5	3	4	6
b_k	15	12	9	16	17

17 ~~8~~

مثال 1

حل طالب

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	15	15	15	15	15	15	15	15	15	15	15
2	0	0	15	15	15	15	15	27	27	27	27	27	27
3	0	0	15	15	15	24	24	27	27	27	36	36	36
4	0	0	15	15	16	24	31	31	31	40	40	43	43
5	0	0	15	15	16	24	31	31	32	40	40	43	48

$V[1,1] = V[0,1]$

$V[1,2] = \max(V[0,2], V[0,2-2]+15) = 15$

$V[1,3] = \max(V[0,3], V[0,3-2]+15 = 15$

$V[1,4] = \max(V[0,4], V[0,4-2]+15 = 15$

$V[1,5] = \max(V[0,5], V[0,5-2]+15 = 15$

$V[1,6] = \max(V[0,6), V[0,6-2]+15$

$V[2,1] =$

$V[2,5] = \max(V[1,5], V[1,5-5]+12$

$V[2,6] = \max(V[1,6], V[1,6-5]+12 =$

$V[2,7] = \max(V[1,7], V[1,7-5]+12 = 15+12 = 27$

$V[2,8]\ \max(V[1,8], V[1,8-5]+12 =$

$V[2,9]\ \max(V[1,9], V[1,9-5]+12 =$

$V[2,10]\ \max(V[1,10], V[1,10-5]+12$

$V[2,11]\ \max(V[1,11], V[1,11-5]+12$

$V[2,12]\ \max(V[1,12], V[1,12-5]+12$

$V[4,11] = V[3,11], V(3,11-4]+16$ 27 +16

$V(3,2) = \max(V[2,3], V[2,3-3]+9 = 15$

$V(3,4) = \max(V[2,4], V[2,4-3]+9$

$V(3,5) = \max(V[2,5], V[2,5-3]+9$

$V(3,6) = \max(V[2,6], [2,6-3]+9$

$V(3,7)\ \max(V[2,7], V[2,7-3]+9$

$V(3,8)\ \max(V(2,8), V(2,8-3)+9$

$V(3,9) = \max(2,9), V(2,9-3)+9$

$V(3,10) = \max(2,10), V(2,10-3)+9$

$V(3,11) = \max(V(2,11), V(2,11-3)+9$

$V(3,12) = \max(V(2,12), V(2,12-3)+9$

$V(4,4) = \max(V[3,4], V[3,4-4]+16$

$V(4,5) = \max(V(3,5], V(3,5-4]+16$

$V[4,6] = \max(V(3,6), V(3,6-4)+16$

$V[4,7] = \max(V(3,7), V(3,7-4)+16$

$V[4,8] = \max(V(3,8), V(3,8-4)+16$

$V[4,9] = \max(V(3,9], V(3,9-4]+16$

$V[4,10] = \max(V[3,10], V(3,10-4]+16$

$\max(V[4,6], V[4,6-6]+17$

$\underset{0}{}$

$[?] = \max(V[4,7], V[4,7-6]+17$

$\underset{1}{}$

$[5,8] = \max(V[4,8], V[4,8-6]+17$

2 15

$V[5,9] = \max(V[4,9], V[4,9-6]+17$

7 32

$V[5,10] = \max(V[4,10], V[4,10-6]+17$

18

30

$V[5,11] = \max(V[4,11], V[4,11-6]+17$

5

$24+17=41$

$V[5,12] = \max(V[4,12], V[4,12-6]+17$

6

31

48